PN - CN1334510 A 20020206
PD - 2002-02-06
PR - CN20010142033 20010907
OPD - 2001-09-07
TI - Remoteboot method of computer in network environment
AB - A remote booting method of computer in network environment includes turning-on the computer, obtaining ID of client computer by initial starting codes in ROM according to dynamic host configuration extension protocol, downloading a script language explanation and execution program from server according to active program transmission protocol, executing the program, choosing operating system by client user, downloading the kernel mirror of operating system and executing the kernel mirror. Its advantages are high reliability, safety and efficiency, and convenient application.
IN - WANG YONG (CN); ZHOU YUEZHI (CN); ZHANG YAOXUE (CN)
PA - UNIV QINGHUA (CN)
EC - G06F9/445B8
IC - G06F9/445 ; G06F13/42

PN - CN1334510 A 20020206 DW200238 G06F9/445 000pp
TI - Remote boot method of computer in network environment
PR - CN20010142033 20010907
PA - (UYQI) UNIV QINGHUA
IC - G06F9/445 ;G06F13/42
IN - WANG Y; ZHANG Y; ZHOU Y
AB - CN1334510 NOVELTY - A remote booting method of computer in network environment includes turning-on the computer, obtaining ID of client computer by initial starting codes in ROM according to dynamic host configuration extension protocol, downloading a script language explanation and execution program from server according to active program transmission protocol, executing the program, choosing operating system by client user, downloading the kernel mirror of operating system and executing the kernel mirror. its advantages are high reliability, safety and efficiency, and convenient application.
- (Dwg.0/0)
OPD - 2001-09-07
AN - 2002-340767 [38]

[19]中华人民共和国国家知识产权局

[51]Int. Cl[7]
G06F 9/445
G06F 13/42

[12]发明专利申请公开说明书

[21]申请号 01142033.2

[43]公开日 2002年2月6日

[11]公开号 CN 1334510A

[22]申请日 2001.9.7 [21]申请号 01142033.2

[71]申请人 清华大学
地址 100084 北京市海淀区清华园
[72]发明人 张尧学 周悦芝 王 勇
彭玉坤 王晓辉

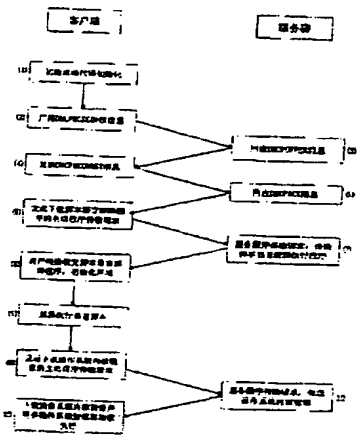
[74]专利代理机构 北京清亦华专利事务所
代理人 廖元秋

权利要求书3页 说明书23页 附图页数5页

[54]发明名称 一种网络环境下的计算机远程启动方法

[57]摘要

本发明属于计算机网络领域，其步骤包括：只读存储器中的初始启动代码在该计 算机加电启动后使用动态主机配置扩展协议获得客户端计算机的标识；然后通过主动 程序传输协议从服务器上下载一个脚本语言解释执行程序并加载执行；在该脚本语言 解释执行的环境下，客户端通过对语言脚本的解释执行让用户选择需要加载的操作系 统；客户端再利用主动程序传输协议从服务器上下载操作系统内核镜像并加载执行，从而实现了计算机的远程启动。本发明具有可靠、安全、高效，使用简单，并且能够 支持多操作系统的启动的特点。本发明可用于网络计算机的远程启动，PC 的多操作系 统远程启动，还可以用于智能家电的远程启动，在信息家电领域有很好的应用前景。



ISSN1003-4274

权 利 要 求 书

1、一种网络环境下计算机远程启动的方法，首先在客户端的计算机主板上或网卡上设置一个只读存储器或可擦可编程只读存储器，远程启动时包括以下步骤：

1） 只读存储器中的初始启动代码在该计算机加电启动后使用动态主机配置扩展协议获得客户端计算机的标识；

2） 然后通过主动程序传输协议从服务器上下载一个脚本语言解释执行程序并加载执行；

3） 在该脚本语言解释执行的环境下，客户端通过对语言脚本的解释执行让用户选择需要加载的操作系统；

4） 客户端再利用主动程序传输协议从服务器上下载该操作系统的内核镜像并加载执行，从而实现了计算机的远程启动。

2、如权利要求 1 所述的网络环境下计算机远程启动的方法，其特征在于，

所说的步骤 1）具体包括如下的步骤：

（1） 对初始启动代码进行初始化：该过程包括检查代码自身的合法性和完整性；初始化本机环境和网络环境等；

（2） 初始启动代码驱动网卡工作，并广播 DHCPDISCOVER 消息分组，请求本机的 IP 地址，网关地址，启动服务器地址，脚本语言解释执行程序等标识；

（3） 服务器接收到 DHCPDISCOVER 消息分组后，检查第 60 号和 43 号选项，如果是动态主机配置扩展协议的消息分组，则向客户端发送应答的 DHCPOFFER 消息分组；如果不是动态主机配置扩展协议的消息分组，则将其丢弃；

（4） 客户端收到 DHCPOFFER 消息分组后，验证该消息分组是否是动态主机配置扩展协议的消息分组，如果该消息分组是动态主机配置扩展协议的消息分组，则记录有关动态主机配置协议参数，服务器地址，脚本语言解释执行程序等内容，否则丢弃该消息分组，同时客户端向服务器发送 DHCPREQUEST 消息分组并等待服务器的同意应答；

（5） 服务器收到 DHCPREQUEST 消息分组后，向客户端发送 DHCPACK 消息分组作为应答。

3、如权利要求 1 所述的网络环境下计算机远程启动的方法，其特征在于，

所说的步骤 2）具体包括如下的步骤：

（1）客户端收到 DHCPACK 消息分组后，验证该消息分组是否是动态主机配置扩展协议的消息分组，如果该消息分组是动态主机配置扩展协议的消息分组，则记录相关的内容，否则丢弃该消息分组；同时客户端根据返回的内容向服务器发出请求下载脚本语言解释执行程序的传输请求；

（2）服务器接收到客户端的程序传输服务的请求后，利用主动程序传输协议的单播或多播方式向客户端传送脚本语言解释执行程序。

4、如权利要求 1 所述的网络环境下计算机远程启动的方法，其特征在于，

所说的步骤 3）具体包括如下的步骤：

（1）客户端接收完脚本语言解释执行程序，并将其存储在本地内存中；客户端加载执行脚本语言解释执行程序，初始启动代码释放系统控制权，脚本语言解释执行程序初始化脚本语言解释执行的环境；

（2）客户端脚本语言解释执行的环境解释执行语言脚本，在此阶段，用户使用键盘选择需要加载的操作系统。

5、如权利要求 1 所述的网络环境下计算机远程启动的方法，其特征在于，所说的步骤 4）具体包括如下的步骤：

（1）客户端根据用户选择的操作系统的内核镜像的程序名称、路径或摘要，向服务器发出请求下载操作系统内核镜像的程序传输请求；

（2）服务器接收到客户端发出的程序传输服务的请求后，利用主动程序传输协议的单播或多播方式向客户端传送操作系统的内核镜像；

（3）客户端接收操作系统内核镜像并将其存储在本地内存中，客户端再利用多操作系统加载器加载执行操作系统内核镜像，至此，操作系统加载完毕。

6、如权利要求 1 所述的网络环境下计算机远程启动的方法，其特征在于，所说的动态主机配置扩展协议是对动态主机配置协议 DHCP 的选项进行自定义而得到的，包括如下的内容：

1）标签号为 60 的选项，该选项的值定义为“MRBM-EDHCPSEClient”；

2）标签号为 43 的选项，该选项的值定义为“03:05:08:00:00:00:ff:ff”；

3）标签号为 165 的选项，该选项的用来传递给脚本语言解释执行程序的参数；

4）标签号为 138 的选项，该选项定义为主动程序传输协议进行传输程序时的多播地址；

5）标签号为 139 的选项，该选项的含义为主动程序传输协议以多播方式传输程序时客户端监听主动程序传输协议数据包的 UDP 端口；

6）标签号为 140 的选项，该选项的含义为主动程序传输协议以多播方式传输程序时服务器监听主动程序传输协议请求的 UDP 端口；

7）标签号为 141 的选项，该选项的含义为客户端重发主动程序传输协议请求前等待的时间；

8）标签号为 142 的选项，该选项的含义为服务器监听主动程序传输协议请求的延迟时间；

9）标签号为 143 的选项，该选项的含义为主动程序传输协议要传输程序的摘要。

7、如权利要求 1 所述的网络环境下计算机远程启动的方法，其特征在于，所说的主动程序传输协议在客户端下载程序时的步骤如下：

1）客户端发出读请求包，该数据包中包含有程序的名称或摘要以及请求的数据块的块号，初始为 1；

2）服务器收到客户端的读请求包后，检查程序是否存在，如果存在，则发送数据包，否则发出错误包；

3）客户端如果收到数据包，则检查数据包中的块号是否存在，如果存在则丢弃该数据包，如果不存在，则保留在缓冲区；如果收到的数据块的大小不足 512 个字节，则转 6），否则将块号加 1，转 1）；

4）客户端如果收到错误的包，则终止传输过程，并提示用户；

5）客户端如果在一定时间后，如果没有收到服务器的应答消息，则重发请求，如果重发次数到达了 10 次，则终止传输过程，并向用户报告；

6）如果传输使用的是摘要的方法，则对程序进行基于摘要的检查，如果程序正常通过检查，则转 8），否则转 7）；

7）表明传输过程出现了错误，需要重新开始传输，如果重新开始的次数超过了 3 次，则终止传输，并向用户报告；否则转 1）；

8）传输正确，正常结束传输过程。

8、如权利要求 1 所述的网络环境下计算机远程启动的方法，其特征在于，所说的操作系统的内核镜像的格式内容具体包括：

1）内核镜像标识，目前的值为 05h, 0ah, 0ah, 05h，作为内核镜像的标识；

2）内核镜像在内存中的装载地址；

3）内核镜像的首次执行地址，当内核镜像加载程序成功地装载到装载地址后，跳

转到执行地址处执行，从而释放系统的控制权；

4）内核镜像的大小，该长度是指内核镜像为解压前的大小；

5）内核镜像解压后所占用的内存大小；

6）程序的真正数据区。

说　明　书

一种网络环境下的计算机远程启动方法

技术领域

本发明属于计算机网络领域，特别涉及网络环境下计算机远程启动的方法。

背景技术

个人计算机的出现，使得计算机变得比过去的大型计算机更加好用，界面友好，从而使计算机走进了人类的日常生活。然而，随着计算机的发展，软件越来越庞大，从而又使计算机的维护工作量非常大。企业和学校这种公共机房的维护管理，尤其是令系统管理员感到头疼的问题。因此，人们一直在寻找一种方法，希望能够减轻维护计算机系统的工作量，使计算机的使用越来越简单，而且又能够降低成本。

计算机网络的出现，使得 PC 机的计算模式发生了变化，由原来的分散的计算，各不相干的独立的计算机系统，发展到了进行相互通信，进行资源共享的网络系统。这为解决上述问题提供了一个技术的基础。

计算机的远程启动就是建立在计算机网络技术基础上的一种计算机的启动方法。其基本的工作原理是首先使系统连上网络，然后通过协议从网络的服务器上下载操作系统软件并加以执行。计算机的远程启动能够大大减轻系统维护工作量，并能降低系统的整体造价。同时，由于操作系统及其他的应用都存储在服务器上，因此使得终端系统非常健壮。

计算机远程启动技术，是指计算机的启动不需要从本地的存储设备上读取操作系统镜像，而是从远程的服务器上下载软件镜像并在本地执行。在远程启动技术的基础上产生了“无盘工作站”以及其他的计算模式（如瘦客户端技术）等，它们由于非常易于维护和管理而受到了人们的青睐。Novell 公司是较早推出远程启动技术的公司之一。它提出了一种用于远程启动的 RPL（Remote Program Load）协议，但是由于使用 RPL 必须用 Novell 组网，其命令繁多，安装配置非常复杂；支持 WIN95 远程引导的 RPL 技术通用性差，技术不成熟，已经几乎被淘汰了。

由于远程启动的种种好处，市场需要新的远程启动技术，尤其是能支持 WIN98 的远程启动技术。在这种要求下，Intel 公司在 1998 年提出了一种称为 PXE（Preboot Execution Environment）的远程启动技术。最新的 PXE 规范的版本是 2.0。PXE 规范提出了一种新的基于 TCP/IP 协议簇的远程启动方法。因为 PXE 是建立在已成为工业标准的 Internet 技术规范的基础之上，所以它的稳定性比较好。但 PXE 技术还未发展成熟，市场尚在成长之中。而且 PXE 技术对多操作系统的启动支持也不够强，它没有提出多操作系统启动的办法。

PXE 协议是一种基于客户/服务器模型的协议，分为客户端和服务器两部分。其启动的基本原理是首先使用动态主机配置协议（DHCP）获得本机的 IP 地址、启动服务器的地址，然后利用简单文件传输协议（TFTP）下载启动程序并执行。

PXE 远程启动的步骤如下：

步骤 1：客户端广播 DHCPDISCOVER 消息分组到标准的 DHCP 端口 67。该消息分组中具有选项域（option field），选项域的内容如下：

1）客户端的唯一标识（Universally Unique ID，UUID）标签（tag）。

2）客户端通用网络设备接口版本标签。

3）客户端系统体系结构标签。

4）DHCP 第 60 号标签，其含义是类别标识（Class ID），其值设置为：

"PXEClient:Arch:xxxx:UNDI:yyyzzz"。

步骤 2：DHCP 或 DHCP 代理服务通过向客户端的标准响应端口 68 发送 DHCPOFFER 消息对客户端进行响应。如果是 DHCP 代理服务器进行响应的话，则客户 IP 地址的值为空（0.0.0.0）。如果是 DHCP 服务器进行响应的话，则客户 IP 地址的值是有效值。

如果客户端没有收到响应，则到了一定时间后，客户端将重新发送广播消息。

步骤 3：客户端从它收到的 DHCPOFFER 消息中，记录以下的信息：

1）DHCP 或 BOOTP 服务提供的客户端 IP 地址以及其他的参数。

2）从 DHCPOFFER 中 PXE 标签的启动服务器域中得到启动服务器列表。

3）如果可能，记录发现控制（Discovery Control）标签的内容。

4）如果可能，记录 Multicast Discovery IP 地址。

步骤 4：如果客户端选择了一个 DHCP 服务器所提供的 IP 地址，则它必须按照标准的 DHCP 协议向 DHCP 服务器发送一个请求并等待服务器的同意应答。如果客户端使用一个 BOOTP 服务器响应，则可以简单地使用这个地址。

步骤 5：客户端选择和发现启动服务器。选择和发现启动服务器的数据分组可能被广播（通过端口 67），可能被多播（通过端口 4011），也可能被单播（通过端口 4011）。这取决于前面所说的 DHCPOFFER 消息的 PXE 扩展标签中发现控制标签的值。这个分组的格式和内容同步骤 1 中所说的 DHPCDISCOVER 消息分组是一样的，只是消息的类型为 DHCPREQUEST。该分组包含如下的内容：

1）DHCP 服务器分配给客户端的 IP 地址。

2）客户端的唯一标识（UUID）标签。

3）客户端通用网络设备接口版本标签。

4）客户端系统体系结构标签。

5）DHCP 第 60 号标签，其含义是产品类别标识（Class ID），其值设置为："PXEClient:Arch:xxxx:UNDI:yyyzzz"。

6）PXE 选项域中的启动服务器类型。

步骤 6：启动服务器利用单播的方式向客户端的源端口发送一个 DHCPACK 应答分组。该分组包含如下的内容：

1）启动文件的名字。

2）MTFTP（Multicast Trivial File Transfer Protocol）配置参数。

3）网络启动程序成功执行需要的其他选项。

步骤 7：客户端下载可执行的文件，该下载可通过标准的 TFTP 协议（端口 69）或 MTFTP 协议（启动服务器应答分组中所分配的端口）。可执行代码在客户端内存中的存放位置取决于客户端 CPU 的体系结构。

步骤 8：PXE 客户端决定是否需要对下载的文件进行授权测试。如果需要测试，则客户端向原来提供启动文件的启动服务器发送另一个 DHCPREQUEST 消息，请求认证文件，通过 TFTP 协议或 MTFTP 下载认证文件，然后执行授权测试。

步骤 9：最后，如果授权测试成功或不需要进行授权测试，则 PXE 客户端启动执行下载的代码。

PXE 协议方法由于采用了标准的 TCP/IP 协议簇，因此比较健壮，也比较容易实现。PXE 的配置和使用要比 RPL 方便得多，但是它具有如下的不足：

其一，安全性。TFTP 协议是一个简单的协议，没有认证机制，因此提供 TFTP 服务的主机可能受到攻击。PXE 提供了一种通过认证文件授权的方法来保证安全。但是如果在 TFTP 传输的过程中程序包被恶意篡改，PXE 方法是没有办法识别出来的。

其二，PXE 主要是通过对 DHCP 的扩展来实现。PXE 不但扩展了 DHCP 的选项定义，也扩展了 DHCP 服务的交互过程，这导致 DHCP 服务器的功能和启动服务器的功能界限

不清，不便于实现，也不便于采用现有的 DHCP 服务来架构远程启动服务。

其三，不能支持多操作系统的启动。PXE 只能让客户端选择为之服务的启动服务器，不能让用户选择他们需要加载的操作系统。

因此，目前流行的计算机远程启动技术，在健壮性，安全性和通用性上不能令人满意，并且也不能对多操作系统的启动提供有效的支持。因此不能满足市场对计算机远程启动技术的要求。

发明内容

本发明针对现有技术的不足之处，提出了一种网络环境下计算机远程启动的方法，具有可靠、安全、高效，使用简单，并且能够支持多操作系统的启动的特点。可用于网络计算机的远程启动、PC 的多操作系统远程启动，还可以用于智能家电的远程启动，在信息家电领域有很好的应用前景。

本发明的方法基于客户/服务器模型的，分为服务器和客户端两部分。这里所说的服务器和客户端是一个服务模型的概念。服务器指提供远程启动服务的计算机，通常称之为启动服务器，而客户端是指需要远程启动的计算机。

本发明提出的一种网络环境下计算机远程启动的方法，首先在客户端的计算机主板上或网卡上设置一个只读存储器（ROM）或可擦可编程只读存储器（EPROM），远程启动时包括以下步骤：

1） 只读存储器中的初始启动代码在该计算机加电启动后使用动态主机配置扩展协议（动态主机配置扩展协议是通过对动态主机配置协议 DHCP 的选项进行自定义而得到的）获得客户端计算机的标识；

2） 然后通过主动程序传输协议从服务器上下载一个脚本语言解释执行程序并加载执行；

3） 在该脚本语言解释执行的环境下，客户端通过对语言脚本的解释执行让用户选择需要加载的操作系统；

4） 客户端再利用主动程序传输协议从服务器上下载操作系统的内核镜像并加载执行，从而实现了计算机的远程启动。

其中 2）和 4）步骤中所使用的传输协议可为已有的任何一种传输协议，比如简单文件传输协议。步骤 3)中的脚本语言，既可为本发明实施方式中提出的脚本语言，也可为已有的其他一种脚本语言。

本发明的网络环境下计算机远程启动方法的详细步骤如图 1 所示，其中：

上述中的步骤 1）具体包括如下的步骤：

(1) 对初始启动代码进行初始化：该过程包括检查代码自身的合法性和完整性；初始化本机环境和网络环境等；

(2) 初始启动代码驱动网卡工作，并广播 DHCPDISCOVER 消息分组，请求本机的 IP 地址，网关地址，启动服务器地址，脚本语言解释执行程序等标识；

(3) 服务器接收到 DHCPDISCOVER 消息分组后，检查第 60 号和 43 号选项，如果是动态主机配置扩展协议的消息分组，则向客户端发送应答的 DHCPOFFER 消息分组；如果不是动态主机配置扩展协议的消息分组，则将其丢弃；

(4) 客户端收到 DHCPOFFER 消息分组后，验证该消息分组是否是动态主机配置扩展协议的消息分组，如果该消息分组是动态主机配置扩展协议的消息分组，则记录有关动态主机配置协议参数，服务器地址，脚本语言解释执行程序等内容，否则丢弃该消息分组，同时客户端向服务器发送 DHCPREQUEST 消息分组并等待服务器的同意应答；

(5) 服务器收到 DHCPREQUEST 消息分组后，向客户端发送 DHCPACK 消息分组作为应答。

上述中的步骤 2）具体包括如下的步骤：

(1)客户端收到 DHCPACK 消息分组后，验证该消息分组是否是动态主机配置扩展协议的消息分组，如果该消息分组是动态主机配置扩展协议的消息分组，则记录相关的内容，否则丢弃该消息分组；同时客户端根据返回的内容向服务器发出请求下载脚本语言解释执行程序的传输请求；

(2) 服务器接收到客户端的程序传输服务的请求后，利用主动程序传输协议的单播或多播方式向客户端传送脚本语言解释执行程序。

上述中的步骤 3）具体包括如下的步骤：

(3) 客户端接收完脚本语言解释执行程序，并将其存储在本地内存中；客户端加载执行脚本语言解释执行程序，初始启动代码释放系统控制权，脚本语言解释执行程序初始化脚本语言解释执行的环境；

(4) 客户端脚本语言解释执行的环境解释执行语言脚本，在此阶段，用户使用键盘选择需要加载的操作系统。

上述中的步骤 4）具体包括如下的步骤：

(1) 客户端根据用户选择的操作系统的内核镜像的程序名称、路径或摘要，向服务器发出请求下载操作系统内核镜像的程序传输请求；

(2) 服务器接收到客户端发出的程序传输服务的请求后，利用主动程序传输协议的单播或多播方式向客户端传送操作系统内核镜像；

(3) 客户端接收操作系统内核镜像并将其存储在本地内存中，客户端再利用多操作系统加载器加载执行操作系统内核镜像，至此，操作系统加载完毕。

上述所说的初始启动代码的组成如图 2 所示，包括如下的四个部分：

1）网卡驱动以及相关协议；

2）动态主机配置扩展协议的客户端代码；

3）主动程序传输协议客户端代码；

4）脚本语言解释执行程序加载器。

初始启动代码启动的过程如图 3 所示，其步骤包括：

1）计算机加电；

2）初始启动代码 ROM 初始化；

3）执行 BIOS INT 19h；

4）初始启动代码检查自身代码的完整性；

5）本地初始化；

6）初始化网络环境。

上述所说的动态主机配置扩展协议是对动态主机配置协议 DHCP 的选项进行自定义而得到的，包括如下的内容：

1）标签号为 60 的选项，该选项的值定义为“MRBM-EDHCPSEClient”；

2）标签号为 43 的选项，该选项的值定义为“03:05:08:00:00:00:ff:ff”；

3）标签号为 165 的选项，该选项的用来传递给脚本语言解释执行程序的参数；

4）标签号为 138 的选项，该选项定义为主动程序传输协议进行传输程序时的多播地址；

5）标签号为 139 的选项，该选项的含义为主动程序传输协议以多播方式传输程序时客户端监听主动程序传输协议数据包的 UDP 端口；

6）标签号为 140 的选项，该选项的含义为主动程序传输协议以多播方式传输程序时服务器监听主动程序传输协议请求的 UDP 端口；

7）标签号为 141 的选项，该选项的含义为客户端重发主动程序传输协议请求前等待的时间；

8）标签号为 142 的选项，该选项的含义为服务器监听主动程序传输协议请求的延迟时间；

9）标签号为 143 的选项，该选项的含义为主动程序传输协议要传输程序的摘要。

上述所说的主动程序传输协议包括如下三种数据包：

1）读请求包；

2）数据应答包；

3）错误包。

主动程序传输协议在客户端下载程序时的步骤如下：

1）客户端发出读请求包，该数据包中包含有程序的名称或摘要以及请求的数据块的块号，初始为 1；

2）服务器收到客户端的读请求包后，检查程序是否存在，如果存在，则发送数据包，否则发出错误包；

3）客户端如果收到数据包，则检查数据包中的块号是否存在，如果存在则丢弃该数据包，如果不存在，则保留在缓冲区。如果收到的数据块的大小不足 512 个字节，则转 6），否则将块号加 1，转 1）；

4）客户端如果收到错误的包，则终止传输过程，并提示用户；

5）客户端如果在一定时间（由 DHCPACK 消息分组中的主动程序传输等待时间选项决定）后，如果没有收到服务器的应答消息，则重发请求，如果重发次数到达了 10 次，则终止传输过程，并向用户报告；

6）如果传输使用的是摘要的方法，则对程序进行基于摘要的检查，如果程序正常通过检查，则转 8），否则转 7）；

7）表明传输过程出现了错误，需要重新开始传输，如果重新开始的次数超过了 3 次，则终止传输，并向用户报告。否则转 1）；

8）传输正确，正常结束传输过程。

上述所说的主动程序传输协议在多播时客户端下载程序的过程，其步骤如下：

1）多播监听；

2）多播打开；

3）多播关闭。

上述所说的操作系统的内核镜像的格式如图 7 所示，内容如下：

1）内核镜像标识，目前的值为 05h, 0ah, 0ah, 05h，作为内核镜像的标识；

2）内核镜像在内存中的装载地址；

3）内核镜像的首次执行地址，当内核镜像加载程序成功地装载到装载地址后，跳转到执行地址处执行，从而释放系统的控制权；

4）内核镜像的大小，该长度是指内核镜像为解压前的大小；

5）内核镜像解压后所占用的内存大小；

6）程序的真正数据区。

本发明的方法的主要技术特点是：

一、在客户端使用一段称为初始启动代码的代码，该代码驻留在主板或网卡设备的 ROM 或 EPROM 中。初始启动代码主要由四部分组成：网卡驱动以及相关的网络协议代码；动态主机配置扩展协议客户端代码；主动程序传输协议客户端代码；脚本语言解释执行程序加载器。服务器为客户端的启动提供服务，主要包括动态主机配置协议服务和主动程序传输协议的程序传输服务。通过客户端和服务器的一系列的交互过程，实现客户端从服务器上下载用户选择的操作系统的内核镜像。

二、使用扩展的动态主机配置协议。为了避免网络上其他动态主机配置协议服务器对本方法提供的动态主机配置协议的影响，本方法利用动态主机配置协议的 60 和 43

标签来标识本方法所使用的动态主机配置协议服务。这使本方法具有可靠性。同时，我们还对动态主机配置协议的 165，138-143 标签进行了定义，这一定义的主要目的是为了使主动程序传输服务能够支持多播，从而减少多台计算机并发启动时对网络流量的冲击。这使本方法具有很强的扩展能力。

三、本方法所使用的主动程序传输协议简单，安全。该传输协议是基于 UDP 协议的，因而实现简单。主动程序传输协议的数字签名技术，保护程序在传输过程中不被恶意篡改，也保证了传输的准确可靠。该协议是一个无状态协议，它使得在传输过程中计算机能够自动进行恢复，非常健壮。同时，主动程序传输协议能够支持多播传输，大大减少了本方法对网络带宽的依赖性。

四、脚本语言解释执行程序的加载执行为脚本语言的解释执行提供了一个类似操作系统的环境。通过对脚本语言的解释执行，能够接受用户的输入，让用户选择操作系统。这使得本方法能够支持多操作系统的远程启动。

五、通过对操作系统内核镜像的格式进行统一规定，本方法能够避开具体操作系统加载在内存分配等方面的不一致。从而达到了能够对多操作系统使用一致的方法来进行加载的效果。

表 1 是本发明的方法与传统的 RPL 远程启动方法以及 PXE 启动方法的比较结果。

表 1 本发明的方法，RPL 和 PXE 的比较表

方法 / 项目	RPL	PXE	本发明的方法
安全性	较差	较差	好
可靠性	较差	较高	高
通用性	较差	好	好
响应时间	短	短	短
支持多播	不支持	支持	支持
实现难度	难	容易	容易
使用情况	使用复杂	使用简单	使用简单
工作效率	较高	较高	较高
启动时间	短	短	稍长
对广域网的支持	无	支持	支持
支持多操作系统	不支持	不支持	支持

综上所述，本发明方法可靠、安全、高效，使用简单，并且能够支持多操作系统的启动。可用于网络计算机的远程启动、PC 的多操作系统远程启动，还可以用于智能家电的远程启动，在信息家电领域有很好的应用前景。

附图说明

图 1 为本发明提出的网络环境下计算机远程启动的流程图。

图 2 为本实施例初始启动代码的结构图。

图 3 为本实施例初始启动代码启动过程的示意图。

图 4 为本实施例主动程序传输协议包格式的示意图。

图 5 为本实施例主动程序传输协议多播监听过程的流程图。

图 6 为本实施例主动程序传输协议多播打开过程的流程图。

图 7 为本发明操作系统内核镜像格式组成的示意图。

具体实施方式

下面结合附图及实施例更加详细地说明本发明的内容。

本发明提出了一种网络环境下计算机远程启动法的实施例。该实施例是基于客户/服务器模型的，分为服务器和客户端两部分。服务器为客户端计算机提供启动服务。

本实施例的具体实现方法包括以下步骤：

1 初始启动代码进行初始化。该过程包括检查代码自身的合法性和完整性；初始化本机环境和网络环境等。

2 初始启动代码驱动网卡工作，并广播 DHCPDISCOVER 消息分组，请求本机的 IP 地址，网关地址，启动服务器地址，脚本语言解释执行程序等。在本实施例中，客户端广播的 DHCPDISCOVER 消息分组中包含如下的内容：

1）第 60 号标签，其含义是产品类别标识（Class ID），设置为："MRBM-EDHCPSEClient"。

2）第 43 号标签，该选项的含义是与产品相关的特定信息，设置为"03:05:08:00:00:00:ff:ff"。

根据以上两个选项域，就可以判断一个动态主机配置协议的消息分组是否是动态主机配置扩展协议的消息分组。

3 服务器接收到 DHCPDISCOVER 消息分组后，检查第 60 号和 43 号选项，如果是动态主机配置扩展协议的消息分组，则向客户端发送应答的 DHCPOFFER 消息分组。如果不是动态主机配置扩展协议的消息分组，则将其丢弃。

4 客户端收到 DHCPOFFER 消息分组后，验证该消息分组是否是动态主机配置扩展协议的消息分组，如果该消息分组是动态主机配置扩展协议的消息分组，则记录有关的信息，否则丢弃该消息分组。其记录的内容包括：

1）动态主机配置服务提供的客户端 IP 地址以及其他参数。

2）服务器的地址。

3）脚本语言解释执行程序的程序名称（如果使用摘要标识的话，则是程序的摘要）。

4）如果有，记录主动程序传输协议多播方式时客户端和服务器所使用的端口。

5）其他必要的参数。

同时客户端向服务器发送 DHCPREQUEST 消息分组并等待服务器的同意应答。

5 服务器收到 DHCPREQUEST 消息分组后，向客户端发送 DHCPACK 消息分组作为应答。该消息分组包含的内容有：

1）动态主机配置协议服务提供的客户端 IP 地址以及其他参数。

2）服务器的地址。

3）脚本语言解释执行程序的程序名称或者摘要。

4）主动程序传输协议多播方式时客户端和服务器所使用的端口。

5）其他参数。

6 客户端收到 DHCPACK 消息分组后，验证该消息分组是否是动态主机配置扩展协议的消息分组，如果该消息分组是动态主机配置扩展协议的消息分组，则记录相关的内容，否则丢弃该消息分组。同时客户端根据返回的内容向服务器发出请求下载脚本语言解释执行程序的传输请求。

7 服务器接收到客户端的程序传输服务的请求后，利用主动程序传输协议的单播或多播方式向客户端传送脚本语言解释执行程序。

8 客户端接收完脚本语言解释执行程序，并将其存储在本地内存中。客户端加载执行脚本语言解释执行程序，初始启动代码释放系统控制权。脚本语言解释执行程序初始化脚本语言解释执行的环境。

9 客户端脚本语言解释执行的环境解释执行语言脚本，在此阶段，用户使用键盘选择需要加载的操作系统。

10 客户端根据用户选择的操作系统的信息，指操作系统镜像的程序名称、路径或摘要，向服务器发出请求下载操作系统内核镜像的程序传输请求。

11 服务器接收到客户端发出的程序传输服务的请求后，利用主动程序传输协议的

单播或多播方式向客户端传送操作系统内核镜像。

12 客户端接收操作系统内核镜像并将其存储在本地内存中。客户端再利用多操作系统加载器加载执行操作系统内核镜像。至此，操作系统加载完毕。

下面对每个步骤的实施进行更详细的说明，并给出相应的实施例。

本实施例的步骤 1 的详细说明：

初始启动代码的搁置方法。

根据现在标准 BIOS 启动规范（BIOS Boot Specification），计算机加电自检通过后，主要搜索下面可用于启动 OS 的设备：

1）BIOS 意识设备（BIOS Aware IPL Devices，BAID），指软盘、硬盘和 SCSI 设备。

2） PnP 卡设备。一般指带有可选 ROM 的网络设备，该方法又可分为 BCV（Boot Connection Vector）方法和 BEV（Bootstrap Entry Vector）方法。BCV 方法利用 INT 13h 钩子程序来实现。BEV 方法通过 PnP 扩展头（PnP Expansion Header）结构中的启动程序入口向量（BEV）来指示 ROM 中的 OS 加载程序。

3）传统卡设备。指带有可选 ROM 的非 PnP 卡设备（例如 ISA 卡设备）。这种方法的可选 ROM 的地址必须在内存统一编址中 C0000h 和 EFFFFh 间 2k 边界处。

本实施例的初始启动代码的搁置利用 BEV 方法来实现，即在 PnP 网卡上增加一个可选 ROM 的办法。在该 ROM 中驻留初始启动代码，计算机加电自检后，将把计算机的控制权交给初始启动代码。但是从原理上说，初始启动代码的驻留和启动也可通过上述中的别的方法来实现。在用于嵌入式设备时，初始启动代码 ROM 也可以安置在主板上。

本实施例初始启动代码的主要组成如图 2 所示，各个部分的功能分别描述如下：

1）网卡驱动以及相关协议。该部分代码的功能主要是初始化网卡，使网卡工作。不同网卡设备该部分代码是不一样的。在本发明的实施例中，网络设备上可选 ROM 中只有本网卡的驱动代码。相关的协议主要是基于 TCP/IP 的协议栈。但是在实施例的初始启动代码中，不是一个完整的 TCP/IP 协议，该协议栈只实现了 IP 层的简单功能（发包和收包）以及 UDP 协议。在实现上也没有严格按照层的概念。所有的这些手段，都是为了能在把代码限制在 32KB 的范围内。

2）动态主机配置扩展协议的客户端代码。该部分的功能主要是利用动态主机配置协议与服务器上的动态主机配置协议服务进行交互，从而获得本机的相关标识以及其他信息。关于动态主机配置扩展协议实施中的具体内容，我们在后面进行详细地说明。

3）主动程序传输协议客户端代码。主动程序传输协议是一个程序传输协议，利用该协议客户端可以从服务器上下载程序到本机。关于主动程序传输协议的具体内容，我们将在后面进行详细地说明。

4）脚本语言解释执行程序加载器。该加载器主要用来脚本语言解释执行程序。由于脚本语言解释执行程序的代码短小，只是一个脚本的解释执行环境，这部分代码的功能比较简单。

初始启动代码 ROM 的头结构。该头结构是 BIOS 与可选 ROM 进行交互的说明信息。标准的可选 ROM 的结构如表 2 所示：

表 2 标准可选 ROM 的头结构

偏移	长	值	描述	备注
0h	2h	AA55	签名	标准
2h	1h	可变	可选 ROM 长度	标准
3h	4h	可变	初始化向量	标准
7h	13	可变	保留	标准
1Ah	2h	可变	扩展头结构偏移量	PnP

为了与标准的可选 ROM 兼容，只对保留区进行了扩展。本实施例中初始启动代码 ROM

的头结构如表 3 所示：

表 3　初始启动代码 ROM 的头结构

偏移量	长度	名称	内容
0h	2h	签名	55h，0AAh
2h	1h	ROM 长度	可选 ROM 长度，以 512 字节为单位
3h	4h	初始化入口	初始化向量
7h	Bh	保留	不定
12h	8h	初始启动代码标识	'MRBMv1.0'
1Ah	2h	可变	扩展头结构偏移量

本实施例初始启动代码的启动过程如图 3 所示。其步骤如下：

1 计算机加电：

计算机加电后将进行自检（Power-On Self Test），即所谓的 POST 过程。BIOS 将对计算机的设备进行初始化，设置中断，初始化输入输出设备，并对设备进行资源的分配等。这一工作完成后，内存的使用情况如下：

表 4　计算机加电 video 初始化后的内存使用情况

基本内存地址	长度	描述
0h	400h	中断向量表
400h	100h	系统 BIOS 数据段
500h	可变	空闲基本内存（部分区域可能被系统 BIOS 使用）
9F800	可变	扩展 BIOS 数据区
高端内存地址	长度	描述
A0000h	20000h	Video RAM（典型情况下）
C0000h	8000h	Video ROM（典型情况下）
C8000h	可变	可选 ROM 和高端内存
E0000h	10000h	系统 BIOS
F0000h	10000h	系统 BIOS
扩展内存	长度	描述
100000h	可变	空闲扩展内存

2 初始启动代码 ROM 初始化：

BIOS 在初始化的过程中将检查到初始启动代码 ROM 的存在，并读取 ROM 中的初始化向量，见表 3。然后 BIOS 将执行初始化过程。该初始化过程将对网卡进行必要的配置。该过程结束后，内存的使用情况如下：

表 5　初始启动代码 ROM 初始化后的内存使用情况

基本内存地址	长度	描述
0h	400h	中断向量表
400h	100h	系统 BIOS 数据段
500h	可变	空闲基本内存（部分区域可能被系统 BIOS 使用）
9F800	可变	扩展 BIOS 数据区
高端内存地址	长度	描述
A0000h	20000h	Video RAM（典型情况下）
C0000h	8000h	Video ROM（典型情况下）
C8000h	8000h	初始启动代码 ROM
D0000h	可变	其他可选 ROM 和高端内存
E0000h	10000h	系统 BIOS
F0000h	10000h	系统 BIOS
扩展内存	长度	描述
100000h	可变	空闲扩展内存

3 执行 BIOS INT 19h：

INT 19h 是 BIOS POST 过程中的最后一步，INT 19h 扫描系统中可以启动操作系统的设备。如果能找到系统的启动设备，INT 19h 并将控制权移交给启动设备上的启动程序。INT 19h 将根据初始启动代码 ROM 头结构中的扩展头结构偏移地址，找到扩展头结构。即插即用设备的扩展头结构如表 6 所示：

表 6 即插即用设备扩展头结构

偏移量	长度	值	描述	备注
0h	4 字节	'$PNP'	签名标识	与设备无关
04h	字节	不定	保留	01h
05h	字节	不定	长度，以 16 字节为单位	与设备无关
06h	字	不定	下一头结构的偏移地址（如果没有则为 0000h）	与设备无关
08h	字节	00h	保留	与设备无关
09h	字节	不定	校验和	与设备无关
0Ah	双字	不定	设备标识	与 PnP 相关
0Eh	字	不定	厂家字符串指针（可选）	与 PnP 相关
10h	字	不定	产品名字字符串指针（可选）	与 PnP 相关
12h	3 字节	不定	设备类型代码	与 PnP 相关
15h	字节	不定	设备指示器	与 PnP 相关
16h	字	不定	BCV 向量，如果没有则为 0000h	与 PnP 相关
18h	字	不定	断开向量（Disconnect Vector），没有则为 0000h	与 PnP 相关
1Ah	字	不定	启动程序入口点向量（BEV），如果没有则为 0000h	与 PnP 相关
1Ch	字	0000h	保留	与 PnP 相关
1Eh	字	不定	静态资源信息向量（Static Resource Information Vector），如果没有则为 0000h	与 PnP 相关

根据扩展头结构中的启动程序入口点，BIOS 跳转到此处执行，到此，初始启动代码掌握了计算机的控制权。

4 初始启动代码检查代码的完整性：

初始启动代码的完整性检查主要是通过对代码结尾的标志码“AA:00:55:00:FF:FF”进行检查，确保代码的完整无误。如果不完整，则该过程需要再次读取初始启动代码或者自动放弃，并将控制权交给 INT 19h。

5 本地初始化：

初始启动代码的初始化。检测计算机的内存模式（实模式还是保护模式）。如果是保护模式，则需要切换到实模式。初始启动代码将设立自己的代码段，数据段，堆栈段。

6 初始化网络环境：

初始启动代码启动网卡开始工作，建立 TCP/IP 协议栈。到此，初始启动代码已经开始工作。

该过程结束后，内存的分配使用情况如下：

表 7 初始启动代码 ROM 启动后的内存使用情况

基本内存地址	长度	描述
0h	400h	中断向量表
400h	100h	系统 BIOS 数据段
500h	可变	空闲基本内存（部分区域可能被系统 BIOS 使用）
95800h	2000h	CPU 堆栈段
97800h	4000h	初始启动代码的数据段

9B800h	4000h	初始启动代码的代码段
9F800h	可变	扩展 BIOS 数据区
高端内存地址	长度	描述
A0000h	20000h	Video RAM（典型情况下）
C0000h	8000h	Video ROM（典型情况下）
C8000h	8000h	初始启动代码 ROM
D0000h	可变	其他可选 ROM 和高端内存
E0000h	10000h	系统 BIOS
F0000h	10000h	系统 BIOS
扩展内存	长度	描述
100000h	可变	空闲扩展内存

本实施例的步骤 2 的详细说明：

步骤 2 中涉及的问题就是本方法是如何扩展定义动态主机配置协议的，即动态主机配置扩展协议的定义和用法。

动态主机配置扩展协议是在 IETF RFC2131 的基础上通过定义动态主机配置协议的消息域以及可选项的含义而得到的。动态主机配置协议能够为 Internet 上的主机提供网络配置参数，诸如计算机的 IP 地址，路由信息，TTL 和最大包长等。动态主机配置协议包括了两个方面的内容，一个是从服务器到主机的配置参数传递协议；另一个是给主机分配网络地址的机制。动态主机配置协议共有两种类型的消息，一种是要求消息（BOOTREQUEST），一种是应答消息（BOOTREPLY）。动态主机配置协议消息分组格式的各个域的长度（以 OCTETS 为单位，8 位比特）和含义如表 8 所示。

表 8 动态主机配置协议消息各个域含义的说明

域名	长度	描述
op	1	消息码，1 为 BOOTREQUEST，2 为 BOOTREPLY
htype	1	硬件地址类型
hlen	1	硬件地址长度
hops	1	客户端将其置为 0，如果使用中继代理，则可使用这一域
xid	4	事务 ID，由客户端选择的随机数，用以表示客户端和主机一次通信过程的标识
secs	2	由客户端填写，其含义为客户端重新发起 DHCP 过程前等待的秒数
flags	2	标志，如果为广播，则第一位为'B'，其他为为 0
ciaddr	4	客户端 IP 地址，只有在客户端处于 BOUND，RENEW 或 REBOUNDINF 状态时才填充
yiaddr	4	客户端 IP 地址
siaddr	4	用于启动过程的下一服务器的地址
giaddr	4	中继代理的 IP 地址
chaddr	16	客户端硬件地址
sname	64	服务器主机名字，以 NULL 结束，可选项
file	128	启动文件名，以 NULL 结束
options	不定	可选的参数域。在 IETF RFC2132 中有已经定义的可选参数的详细说明。

在本实施例中，利用动态主机配置协议来配置客户端的主机参数，并利用 file 域传递需要下载的脚本语言解释执行程序。

当网络上存在多个动态主机配置协议服务器的时候，根据动态主机配置协议的规定，客户端将随机选择一个服务器为其服务。在本实施例中，这种情况将导致混乱。为了增强本方法的健壮性和安全性，本实施例对动态主机配置协议的可选参数域进行了扩展定义。同时，为了满足主动程序传输协议多播的要求，本实施例也扩展定义了其他一些标签。本实施例对动态主机配置协议的扩展定义见表 9。根据 IETF RFC2132 的规定 69 以后的标签都为未定义的标签。

表 9 动态主机配置扩展协议对选项域的扩展定义

标签名	标 签	长度	描述

	号		
商家标识	60	18	该选项用来表明该消息分组是动态主机配置扩展协议的消息分组，其值必须为“MRBM-EDHCPSEClient”。
商家相关信息	43	8	服务相关信息，动态主机配置扩展协议的定义为“03:05:08:00:00:00:ff:ff”。使用 60 和 43 两个标签，本方法能够避免客户端对来自非动态主机配置扩展协议的消息分组进行响应。
消息类型	53	1	消息类型，1=DHCPDISCOVER，2=DHCPOFFER，3=DHCPREQUEST，4=DHCPDECLINE，5=DHCPACK，6=DHCPNAK，7=DHCPRELEASE，8=DHCPINFORM
脚本语言解释执行程序参数	165	不定	指传递给脚本语言解释执行程序的参数。
主动程序传输协议多播地址	138	4	主动程序传输协议服务器传送程序时的多播地址。
主动程序传输协议客户端端口	139	2	使用多播方式时主动程序传输协议客户端监听应答的 UDP 端口。
主动程序传输协议服务器端口	140	2	使用多播方式时主动程序传输协议服务器监听请求的 UDP 端口。
主动程序传输协议等待时间	141	2	主动程序传输协议客户端重发请求前等待的时间，以秒为单位。
主动程序传输协议监听时延	142	2	主动程序传输协议服务器监听请求的延迟时间，以秒为单位。
程序摘要	143	不定	下一步主动程序传输协议需要传输的程序的摘要。该摘要使用 MD5 单向哈希函数生成的，能保证程序正确无误的传输，也能防止程序在传输的过程中被恶意篡改。

DHCPDISCOVER 消息分组的是由客户端向标准动态主机配置协议服务器 UDP 端口 67 广播发送的。DHCPDISCOVER 分组的内容必须按照 RFC 2131 的规定，并加上本发明的扩展内容。其格式和内容如表 10 所示。

表 10 DHCPDISCOVER 消息分组的格式及内容

动态主机配置协议头		
域（长度）	值	备注
op(1)	1	BOOTP REQUEST 操作码
Htype(1)	*	
hlen(1)	*	
hops(1)	*	
xid(4)	*	
secs(2)	*	
flags(2)	*	
ciaddr(4)	0.0.0.0	客户端必须置为 0.0.0.0
yiaddr(4)	*	客户端端的 IP 地址，由服务器提供
siaddr(4)	*	下一步提供服务的服务器 IP 地址
giaddr(4)	*	
chaddr(16)	xx-xx-xx-xx-xx-xx-xx-xx	客户端 MAC 地址
sname(64)	*	可以被选项中的 66 标签覆盖
bootfile(128)	*	可以被选项中的 67 标签覆盖
99.130.83.99（Magic Cookie，动态主机配置协议可选项域的起始标志）		
动态主机配置协议选项		

标签名	标签号	长度	说明
消息类型	53	1	1=DHCPDISCOVER
消息长度	57	2	最大的动态主机配置协议消息分组长度
商家标识	60	18	该选项用来表明该消息分组是动态主机配置扩展协议的消息分组，其值必须为“MRBM-EDHCPSEClient”

商家相关信息	43	8	值为“03:05:08:00:00:00:ff:ff”。

当客户端发出 DHCPDISCOVER 消息后，客户端必须准备接收响应。

本实施例的步骤 3 的详细说明：

服务器收到 DHCPDISCOVER 消息分组后，将发出 DHCPOFFER 响应消息分组。DHCPOFFER 消息分组的格式和内容也必须遵循 RFC 2131 和 RFC 2132 的有关规定。DHCPOFFER 消息分组的内容如表 11 所示。

表 11 DHCPOFFER 消息分组的格式及内容

动态主机配置协议头		
域（长度）	值	备注
op (1)	2	BOOTP REPLY 操作码
htype(1)	*	
hlen(1)	*	
hops(1)	*	
xid(4)	*	
secs(2)	*	
flags(2)	*	
ciaddr(4)	0.0.0.0	服务器总是置为 0.0.0.0
yiaddr(4)	a0, a1, a2, a3	客户端的 IP 地址，由服务器提供
siaddr(4)	a0, a1, a2, a3	下一步启动服务器的服务器 IP 地址
Giaddr(4)	*	
Chaddr(16)	*	客户端 MAC 地址
Sname(64)	*	可以被选项中的 66 标签覆盖
Bootfile(128)	*	可以被选项中的 67 标签覆盖

99.130.83.99			
动态主机配置协议选项			
标签名	标签号	长度	说明
消息类型	53	1	2=DHCPOFFER
消息长度	57	2	最大的动态主机配置协议消息分组长度
商家标识	60	18	该选项用来表明该消息分组是动态主机配置扩展协议的消息分组，其值必须为“MRBM-EDHCPSEClient”。
商家相关信息	43	8	值为“03:05:08:00:00:00:ff:ff”。
脚本语言解释执行程序参数	165	不定	指传递给脚本语言解释执行程序的参数。
主动程序传输协议多播地址	138	4	主动程序传输协议服务器传送文件时的多播地址。当不使用主动程序传输协议的多播功能时，该标签不使用。
主动程序传输协议客户端端口	139	2	使用多播方式时主动程序传输协议客户端监听应答的 UDP 端口。
主动程序传输协议服务器端口	140	2	使用多播方式时主动程序传输协议服务器监听请求的 UDP 端口。
主动程序传输协议等待时间	141	2	主动程序传输协议客户端重发请求前等待的时间，以秒为单位。
主动程序传输协议监听时延	142	2	使用多播方式时主动程序传输协议服务器监听请求的延迟时间，以秒为单位。当不使用主动程序传输协议的多播功能时，该标签不使用。
程序摘要	143	不定	下一步主动程序传输协议需要传输的程序的摘要。如果使用了 bootfile 域，则该域为空。

DHCPOFFER 消息分组发出后，如果客户端没有收到，则客户端将根据超时情况重新发送 DHCPDISCOVER 消息分组。客户端将重新发送 DHCPDISCOVER 消息分组四次，等待的超时时间分别为 4，8，16 和 32 秒。如果四次发送后还未接收到 DHCPOFFER 响应消

息分组，客户端将放弃尝试，并报告用户。

本实施例的步骤 4 的详细说明：

客户端在接收到 DHCPOFFER 消息分组以后，验证该消息分组是否是动态主机配置扩展协议的消息分组，如果不是则丢弃该分组。否则必须记录 DHCPOFFER 消息分组所提供的所有相关的信息，其内容包括：

1）动态主机配置服务提供的客户端 IP 地址以及其他参数。

2）服务器的地址。

3）脚本语言解释执行程序的程序名称（如果使用摘要标识的话，则是程序的摘要）。

4）如果有，记录主动程序传输协议多播方式时客户端和服务器所使用的端口。

5）其他必要的参数。

同时，客户端必须发送 DHCPREQUEST 消息分组请求确认。DHCPREQUEST 消息分组的格式和内容如表 12 所示。

表 12 DHCPREQUEST 消息分组的格式及内容

动态主机配置协议头			
域（长度）	值		备注
op(1)	1		BOOTP REQUEST 操作码
htype(1)	*		
hlen(1)	*		
hops(1)	*		
xid(4)	*		
secs(2)	*		
flags(2)	*		
ciaddr(4)	a0, a1, a2, a3		服务器总是置为 0.0.0.0
yiaddr(4)	0.0.0.0		客户端端的 IP 地址，由服务器提供
siaddr(4)	0.0.0.0		下一步启动服务器的服务器 IP 地址
giaddr(4)	0.0.0.0		
chaddr(16)	xx-xx-xx-xx-xx-xx-xx-xx		客户端 MAC 地址
sname(64)	*		可以被选项中的 66 标签覆盖
bootfile(128)	*		可以被选项中的 67 标签覆盖
99.130.83.99			
动态主机配置协议选项			
标签名	标签号	长度	说明
消息类型	53	1	3=DHCPREQUEST
消息长度	57	2	最大的动态主机配置协议消息分组长度
商家标识	60	18	该选项用来表明该消息分组是动态主机配置扩展协议的消息分组，其值必须为“MRBM-EDHCPSEClient”。
商家相关信息	43	8	值为“03:05:08:00:00:00:ff:ff”。

本实施例的步骤 5 的详细说明：

服务器在收到客户端的 DHCPREQUEST 消息分组后，更新本地的 IP 地址分配数据库，并用 DHCPACK 消息分组作为对 DHCPREQUEST 消息分组的应答。在 DHCPREQUEST 消息分组中，包含的内容有：

1）动态主机配置协议服务提供的客户端 IP 地址以及其他参数。

2）服务器的地址。

3）脚本语言解释执行程序的程序名称或者摘要。

4）主动程序传输协议多播方式时客户端和服务器所使用的端口。

5）其他参数。

DHCPACK 消息分组的格式和内容如表 13 所示。

表 13 DHCPRACK 消息分组的格式及内容

动态主机配置协议头		
域（长度）	值	备注
op(1)	2	BOOTP REPLY 操作码
htype(1)	*	
hlen (1)	*	
hops(1)	*	
xid(4)	*	
secs(2)	*	
flags(2)	*	
ciaddr(4)	0.0.0.0	服务器总是置为 0.0.0.0
yiaddr(4)	A0, a1, a2, a3	客户端端的 IP 地址，由服务器提供
siaddr(4)	A0, a1, a2, a3	下一步启动服务器的服务器 IP 地址
giaddr(4)	*	
chaddr(16)	*	客户端 MAC 地址
sname(64)	*	可以被选项中的 66 标签覆盖
bootfile(128)	启动文件名称	可以被选项中的 67 标签覆盖
99.130.83.99		
动态主机配置协议选项		

标签名	标签号	长度	说明
消息类型	53	1	3=DHCPREQUEST
消息长度	57	2	最大的动态主机配置协议消息分组长度
商家标识	60	18	该选项用来表明 DHCP 消息是 EDHCP 消息，其值必须为“MRBM-EDHCPSEClient”。
商家相关信息	43	8	值为“03:05:08:00:00:00:ff:ff”。
脚本语言解释执行程序参数	165	不定	指传递给脚本语言解释执行程序的参数。
主动程序传输，协议多播地址	138	4	主动程序传输协议服务器传送文件时的多播地址。当不使用主动程序传输协议的多播功能时，该标签不使用。
主动程序传输协议客户端端口	139	2	使用多播方式时主动程序传输协议客户端监听应答的 UDP 端口。
主动程序传输协议服务器端口	140	2	使用多播方式时主动程序传输协议服务器监听请求的 UDP 端口。
主动程序传输协议等待时间	141	2	主动程序传输协议客户端重发请求前等待的时间，以秒为单位。
主动程序传输协议监听时延	142	2	使用多播方式时主动程序传输协议服务器监听请求的延迟时间，以秒为单位。当不使用主动程序传输协议的多播功能时，该标签不使用。
程序摘要	143	不定	下一步主动程序传输协议需要传输的程序的摘要。如果使用了 bootfile 域，则该域为空。

本实施例的步骤 6 的详细说明：

客户端接收到 DHCPACK 消息分组以后，将确定本机可以使用的 IP 地址信息。并记录其他的信息，可参见表 13。其内容包括：

1）动态主机配置服务提供的客户端 IP 地址以及其他参数。

2）服务器的地址。

3）脚本语言解释执行程序的名称或者摘要。

4）如果有，记录主动程序传输协议多播时客户端和服务器所使用的端口。

5）其他必要的参数。

多播接收阶段：

客户端在这个阶段，接收从服务器上发出的多播数据包。

多播关闭阶段：

当客户端接收到所有数据包后，则客户端完成了所有的传输工作。进入下一步的工作。

本实施例中的主动程序传输协议服务一直监听端口 1051。传输服务服务器的工作步骤大体如下：

1 监听 UDP 1051 端口。

2 接收到主动程序传输协议读请求，判断需要传输的程序是否存在，如果不存在，则转 10。

3 判断是否使用多播，如果使用则转 7。

4 计算所需要传送的数据块。

5 如果系统中的线程数小于 50，则新建立一线程。如果大于 50，则选取一个老线程。

6 随机取一大于 1024 的未使用端口，使用新的线程或选取的老线程使用该端口向客户端发送数据。转 1。

7 如果是使用多播传送数据，则首先发送第一块数据。

8 增大数据块的编号，继续发送多播数据。

9 如果程序数据全部多播结束，则终止多播传输。转 1。

10 发送错误包。转 1。

本实施例的步骤 8 的详细说明：

脚本语言解释执行程序下载后将被解开放置到内存地址为 0x1000:0000 的地方。此过程结束后，内存的使用情况如下：

表 14 脚本语言解释执行程序下载后内存的使用情况

基本内存地址	长度	描述
0h	400h	中断向量表
400h	100h	系统 BIOS 数据段
500h	EB00h	空闲基本内存（部分区域可能被系统 BIOS 使用）
10000h	4000h	脚本语言解释执行程序
14000h	不定	空闲基本内存
97000h	800h	CPU 堆栈段
97800h	4000h	初始启动代码的数据段
9B800h	4000h	初始启动代码的代码段
9F800h	可变	扩展 BIOS 数据区
高端内存地址	长度	描述
A0000h	20000h	Video RAM（典型情况下）
C0000h	8000h	Video ROM（典型情况下）
C8000h	8000h	初始启动代码可选 ROM
D0000h	可变	其他可选 ROM 和高端内存
E0000h	10000h	系统 BIOS
F0000h	10000h	系统 BIOS
扩展内存	长度	描述
100000h	可变	空闲扩展内存

本实施例的步骤 9 的详细说明：

脚本语言解释执行程序执行，取得系统控制权。此时的内存使用情况如下：

表 15 脚本语言解释执行的环境加载后内存的使用情况

基本内存地址	长度	描述

0h	400h	中断向量表
400h	100h	系统 BIOS 数据段
500h	92F00h	空闲基本内存（部分区域可能被系统 BIOS 使用）
93400h	400h	脚本语言解释执行环境的堆栈段
93800h	4000h	脚本语言解释执行环境的代码段
97800h	4000h	脚本语言解释执行环境的数据段
9B800h	4000h	初始启动代码的代码段
9F800h	可变	扩展 BIOS 数据区
高端内存地址	长度	描述
A0000h	20000h	Video RAM（典型情况下）
C0000h	8000h	Video ROM（典型情况下）
C8000h	8000h	初始启动代码可选 ROM
D0000h	可变	其他可选 ROM 和高端内存
E0000h	10000h	系统 BIOS
F0000h	10000h	系统 BIOS
扩展内存	长度	描述
100000h	可变	空闲扩展内存

初始启动代码的代码段的存在是为了复用部分的代码，比如网卡驱动和 TCP/IP 协议栈。当然在实现中，我们也可以不考虑使用初始启动代码，而直接利用脚本语言解释执行程序来进行网络环境的初始化，但必须继承初始启动代码所获得的主机配置信息。

脚本语言解释执行的环境通过对脚本的解释执行，接受用户的输入。根据用户的选择，客户端利用主动程序传输协议下载操作系统内核镜像。

在本实施例中，给出了一个脚本语言的例子。其主要功能是接受用户的输入，并根据用户的选择加载相应的操作系统内核镜像。

该脚本语言是基于命令行的，命令后面是命令的参数，其格式为：

Command parameter 1, parameter 2 ……

下面是脚本语言的语言格式。

语法规则：

1 命令以行为单位进行解析，行与行之间用换行或回车隔离

2 行的最大长度为 255 个字符

3 关键字或变量大小写不敏感

4 “为字符串的分隔符

5 $开始的字符串表示变量

6 转义符的使用：

\b 后退

\n 换行

\r 回车

\t Tab

\nnn 用值为八进制数 nnn 的 ASCII 码代替

\X 表示不在上述的字符中的字符 X

字符串表达式

字符串是指以”开始和以”结束的一串字符。如：

“Hello World!”

数值表达式

本脚本语言只使用 32bitd 的 10 进制整数，从-2,147,483,646 到 2,147,483,647。数值表达式使用

数值表达式中可以包括正数和负数。

表达式的形式为：expr1 op expr2，其中 op 的取值可以为：+，-，*，/，%；为了简单起见，各个运算符之间没有优先级的区别，用括号表示运算的顺序。

例如：((4*6)+8)

延迟命令

delay 命令可用于延迟，参数以秒为单位。

例如：

delay 3　　延迟 3 秒

delay 0.5　　延迟 0.5 秒

文件的命名

文件名为字符串。它们必须用双引号括起来，文件名是大小写敏感的，可用转义符来表示特殊的字符。

如果文件名不是以/开始，则默认为从/tftpboot 目录下寻找。

如果文件名中出现冒号，则冒号前的部分将被解释为计算机名。

例如：

"ald.image"　　表示在/tftpboot 目录下的文件

"/tftpboot/ald.image"　　指绝对路径在/tftpboot 下的文件名为 ald.image 的文件。

"192.168.0.72:/tftpboot/ald.image"　　表示在 IP 地址为 192.168.0.72 计算机/tftpboot/目录下的 ald.image 文件。

监控命令

Echo:

该命令的格式为：Echo "text"，其用途是在屏幕上回现字符串"text"

Beep:

该命令将使计算机发出声。

控制命令

Goto label:

指跳转到标号为 label 的地方解释执行。

If ... :

在本脚本语言中 If 命令只是检查表达式的值是否为真，为真则执行后面的命令，否则就继续往下执行。

例如：

```
If  $val="ERROR"  Turnon
  Beep
    ...
```

上面的脚本表示当变量 val 的值为"ERROR"时，则重启计算机，否则使计算机发声，并继续往下执行。

Set:

设置变量的值。

例如：

```
Set  Val1="sopca computer"
```

Poweroff:

关闭计算机。

Turnon:

重启计算机。

与键盘相关的命令

GetKey(var):
等待用户输入一个键值，并存储在变量 var 中。
WaitForKey duration:
在用户按任意键前延迟 duration 秒。
Input (var):
从键盘读入一以回车结束的字符串。目前只支持 30 个字符。
文本输出命令
Print "text":
在当前屏幕出打印字符串
内核镜像加载命令
LoadImage "kernel image name":
加载内核镜像.
下面是一个用上下箭头选择的菜单脚本例子。两个菜单分别启动 windows98 和 Linux 系统。

```
Set MenuNum=2
Set MenuItem={"Starting Windows98" ,"Starting Linux"}
Set CurrentItem=1
:menu
Set exitemenu=""
Print  "Boot selection"
Set i=1
:loop
Set j=($i –1 )
Print "$MenuItem"{$j}
Print "\r\n"
Set I=($I+1)
If $I <=$MenuNum goto loop
GetKey Key
# 向上键
If "$key"=="" if $CurrentItem>1 Set Current=($CurrentItem– 1)
#向下键
If "$key"=="" if $CurrentItem<$MenuNum Set Current=($CurrentItem+1)
#回车
if "$key"=="\r" Set ExitMenu="OK"
if "$ExitMenu"!= "OK" goto menu
if $CurrentItem ==1 goto a1
if $CurrentItem ==2 goto a2
:a1
LoadImage   "192.168.0.72:/tftpboot/WIN$SOP.SYS"
:a2
LoadImage   "192.168.0.72:/tftpboot/Linux.image"
```

本实施例的步骤 10 的详细说明：

客户端根据用户的选择，向服务器（主动程序传输服务）发出下载操作系统内核镜像的传输请求。

本实施例的步骤 11 的详细说明：

服务器（主动程序传输服务）响应客户端的下载操作系统内核镜像的请求，传输操作系统内核镜像。

本实施例的步骤 12 的详细说明：

如果操作系统的内核镜像不大，则客户端利用主动程序传输协议下载操作系统内核镜像后，内存的使用情况如表 16 所示：

表 16　操作系统内核镜像下载后内存的使用情况（小镜像）

基本内存地址	长度	描述
0h	400h	中断向量表
400h	100h	系统 BIOS 数据段
500h	FB00	空闲基本内存（部分区域可能被系统 BIOS 使用）
10000h	不定	空闲内存
93400h	400h	脚本语言解释执行环境的堆栈段
93800h	4000h	脚本语言解释执行环境的代码段
97800h	4000h	脚本语言解释执行环境的数据段
9B800h	4000h	初始启动代码的代码段
9F800h	可变	扩展 BIOS 数据区
高端内存地址	长度	描述
A0000h	20000h	Video RAM（典型情况下）
C0000h	8000h	Video ROM（典型情况下）
C8000h	8000h	初始启动代码可选 ROM
D0000h	可变	其他可选 ROM 和高端内存
E0000h	10000h	系统 BIOS
F0000h	10000h	系统 BIOS
扩展内存	长度	描述
300000h	可变	空闲扩展内存

如果操作系统的内核镜像比较大，则客户端利用主动程序传输协议下载操作系统的内核镜像后，内存的使用情况如 17 所示：

表 17　操作系统内核镜像下载后内存的使用情况（大镜像）

基本内存地址	长度	描述
0h	400h	中断向量表
400h	100h	系统 BIOS 数据段
500h	不定	空闲基本内存（部分区域可能被系统 BIOS 使用）
93400h	400h	脚本语言解释执行环境的堆栈段
93800h	4000h	脚本语言解释执行环境的码段
97800h	4000h	脚本语言解释执行环境的数据段
9B800h	4000h	初始启动代码的代码段
9F800h	可变	扩展 BIOS 数据区
高端内存地址	长度	描述
A0000h	20000h	Video RAM（典型情况下）
C0000h	8000h	Video ROM（典型情况下）
C8000h	8000h	初始启动代码可选 ROM
D0000h	可变	其他可选 ROM 和高端内存
E0000h	10000h	系统 BIOS
F0000h	10000h	系统 BIOS
扩展内存	长度	描述
100000h	200000h	操作系统内核镜像
300000h	可变	空闲扩展内存

当操作系统内核镜像下载结束后，系统的控制权将交给多操作系统加载器。多操

作系统加载器将根据内核镜像的格式，加载操作系统内核镜像。

为了加载多操作系统的内核镜像，必须对内核镜像的格式加以规定，使用统一的内核镜像格式，这样加载器就能透明地加载各种操作系统的内核镜像。在 Linux 中，加载器工作在实模式下，操作系统内核镜像的大小不能超过 1M。这在嵌入式系统中是完全满足需要的。但是在加载象 Windows 系统时就行不通了。这时需要先加载 DOS 内核，再在 DOS 环境中利用 DOS 工具启动 Windows 系统。但是对 Linux 的加载器而言，它不需要知道这些细节，它只是按照内核镜像的格式要求，在实模式下将内核镜像放置在规定的内存物理区域。然后多操作系统加载器跳转到该处执行，从而释放系统的控制权。

本方法中操作系统的内核镜像的格式如图 7 所示，有关的含义说明如下：

内核镜像标识。目前的值为 05h, 0ah, 0ah, 05h，作为内核镜像的标识。

内核镜像在内存中的装载地址。

内核镜像的首次执行地址。当内核镜像被加载程序成功地装载到装载地址后，加载程序将跳转到执行地址处执行，从而释放系统的控制权。

内核镜像的大小。该大小是指内核镜像解压前的大小。

内核镜像解压后所占用的内存大小。该大小是内核镜像解压后内核所占用的内存的大小。

程序的真正数据区。指内核镜像使用的数据区的大小。

至此，本发明所有的关于网络环境下计算机远程启动方法的实施步骤都已经进行了详细的说明，本实施例仅给出本发明的一种具体实现方法，当不能限定本发明的保护范围，凡对本实施例中所述方法的任何等同变换，均应属于本发明权利要求书中所述的保护范围。

说 明 书 附 图

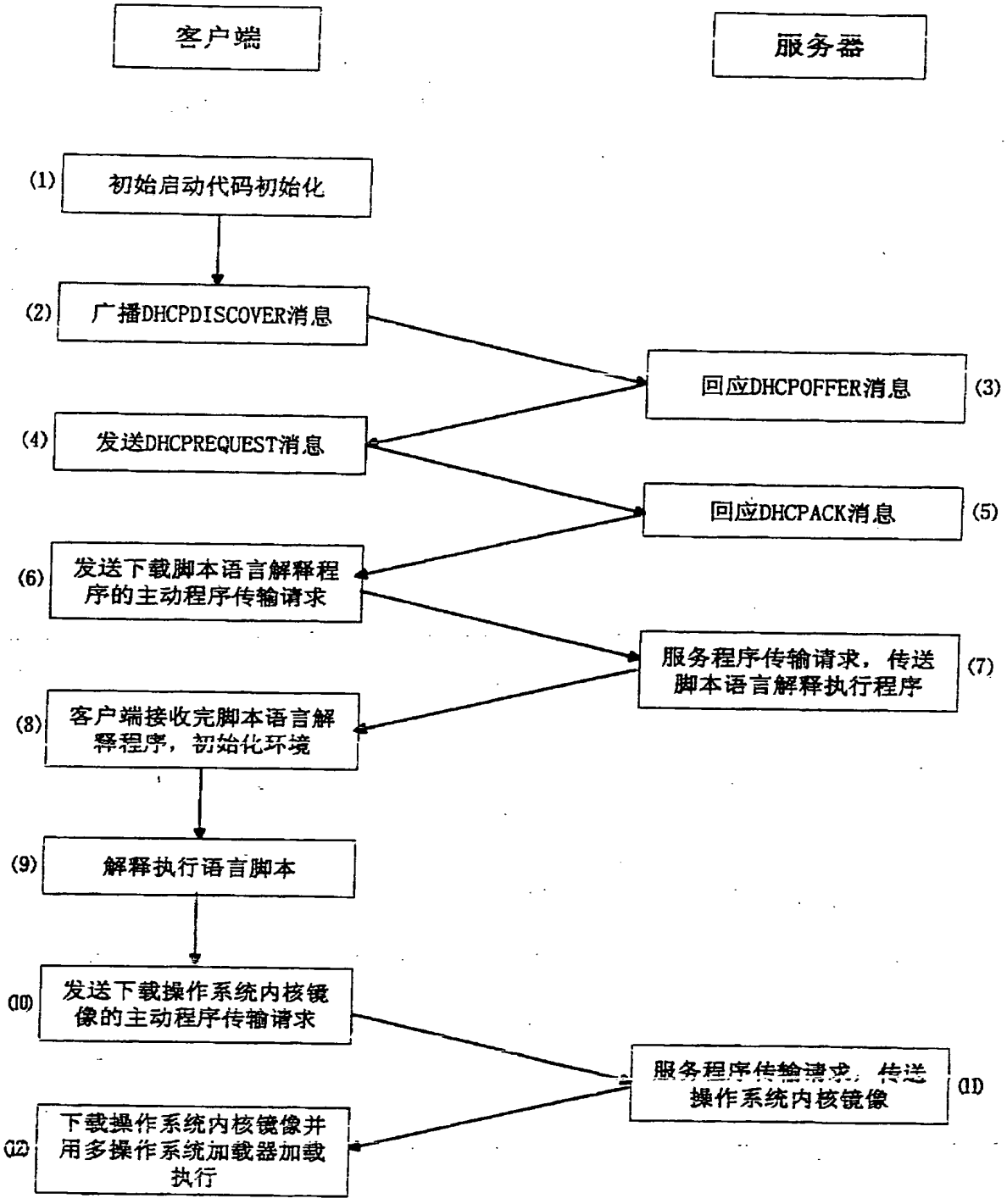

客户端
服务器
(1) 初始启动代码初始化
(2) 广播DHCPDISCOVER消息
回应DHCPOFFER消息 (3)
(4) 发送DHCPREQUEST消息
回应DHCPACK消息 (5)
(6) 发送下载脚本语言解释程序的主动程序传输请求
服务程序传输请求，传送脚本语言解释执行程序 (7)
(8) 客户端接收完脚本语言解释程序，初始化环境
(9) 解释执行语言脚本
(10) 发送下载操作系统内核镜像的主动程序传输请求
服务程序传输请求，传送操作系统内核镜像 (11)
(12) 下载操作系统内核镜像并用多操作系统加载器加载执行

图 1

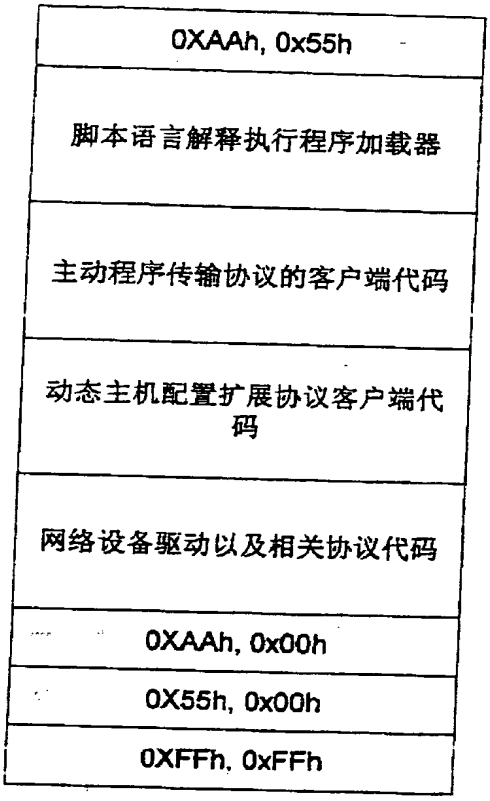

0XAAh, 0x55h
脚本语言解释执行程序加载器
主动程序传输协议的客户端代码
动态主机配置扩展协议客户端代码
网络设备驱动以及相关协议代码
0XAAh, 0x00h
0X55h, 0x00h
0XFFh, 0xFFh

图 2

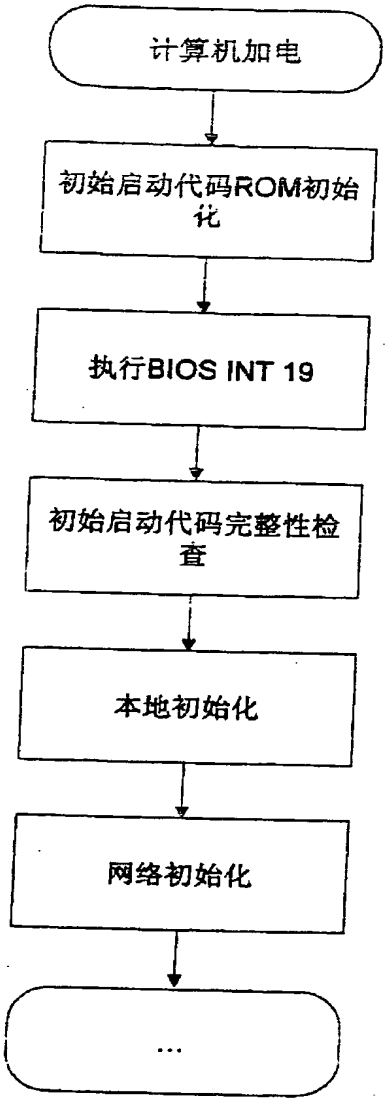

计算机加电
初始启动代码ROM初始化
执行BIOS INT 19
初始启动代码完整性检查
本地初始化
网络初始化
...

图 3

操作码=1 (2)	类型 (1)	程序名称 / MD5 摘要 (字符串)	0 (1)	可选项1	0 (1)	值1	0 (1)	...

读请求包（单播方式）

操作码=1 (2)	类型 (1)	程序名称 / MD5 摘要 (字符串)	0 (1)	MUL	0 (1)	0 (1)	...

读请求包（多播方式）

操作码=2 (2)	类型 (1)	程序名称 / MD5 摘要 (字符串)	0 (1)	数据块号 (2)	数据 (n)

数据应答包

操作码=4 (2)	类型 (1)	错误代码 (2)	错误信息 (字符串)	0 (1)

错误包

图 4

内核镜像标识	4 字节
内核镜像装载地址	双字 格式为 ds:bx
内核镜像首次执行地址	双字 格式为 cs:ip
内核镜像的大小	双字
内核镜像解压后所占用的大小	双字
程序的数据区	可变

图 7

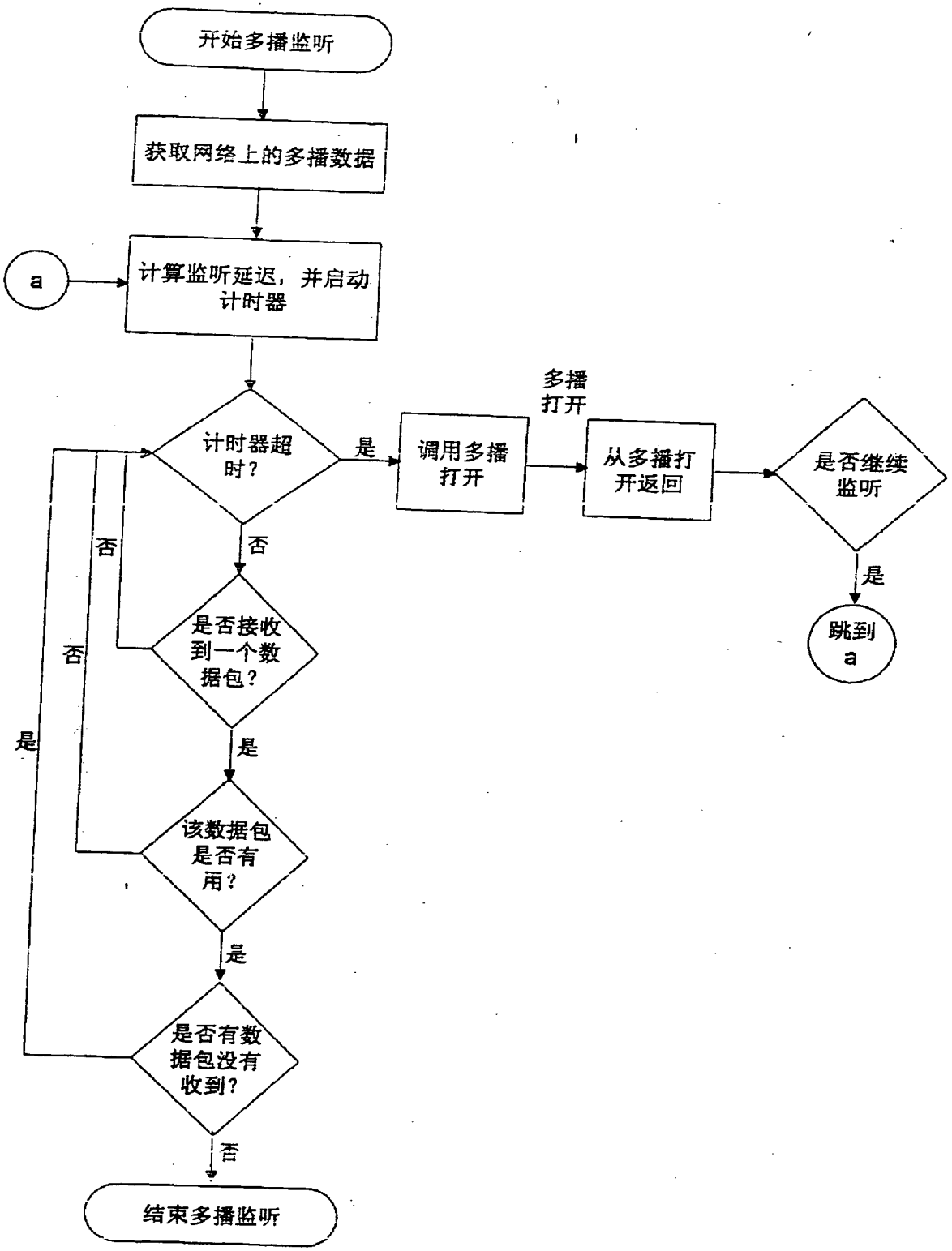

开始多播监听
获取网络上的多播数据
a
计算监听延迟，并启动计时器
计时器超时？
是
调用多播打开
多播打开
从多播打开返回
是否继续监听
是
跳到a
否
否
是否接收到一个数据包？
否
是
该数据包是否有用？
是
是否有数据包没有收到？
是
否
结束多播监听

图 5

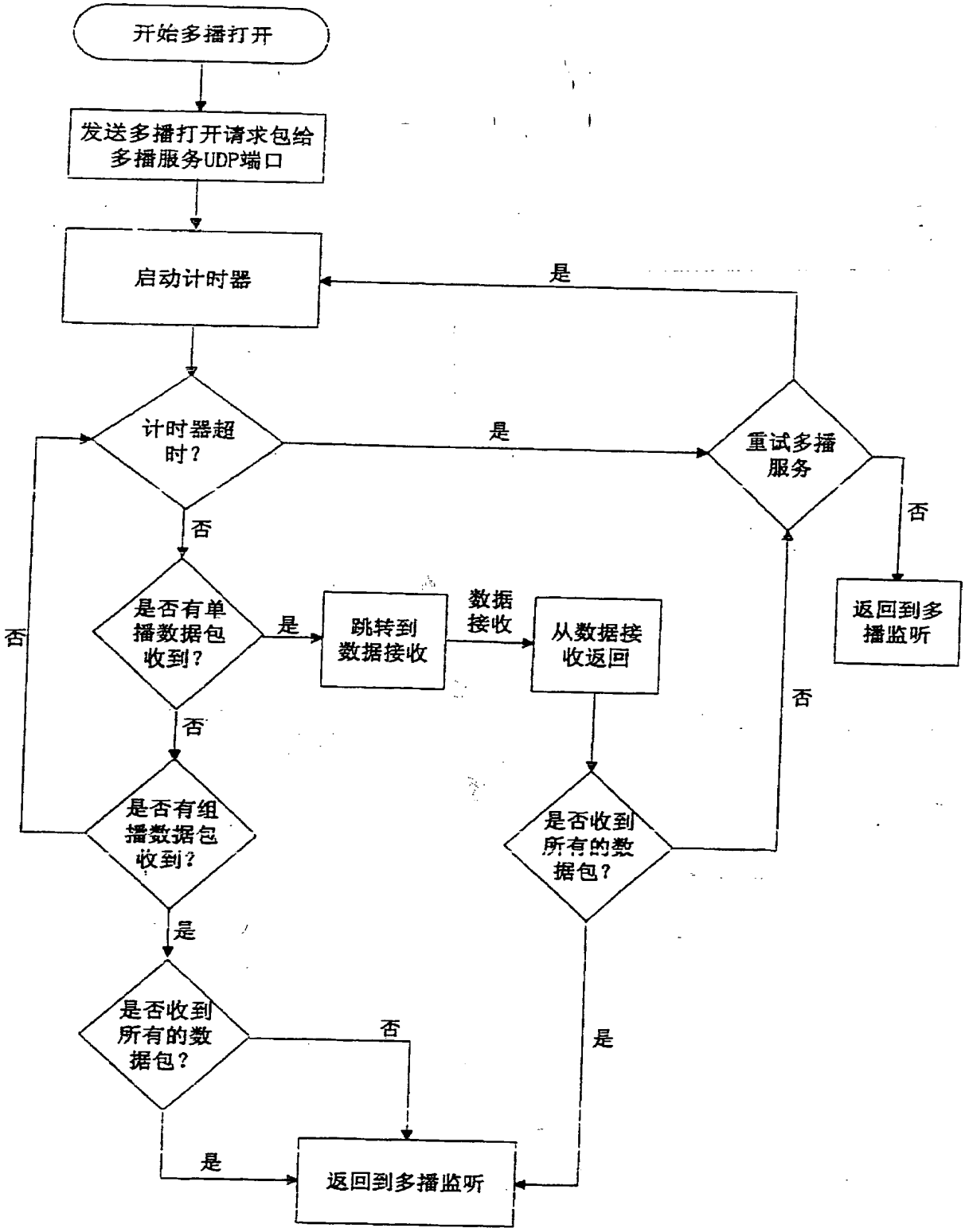

开始多播打开
发送多播打开请求包给多播服务UDP端口
启动计时器
是
计时器超时？
是
重试多播服务
否
返回到多播监听
否
是否有单播数据包收到？
是
跳转到数据接收
数据接收
从数据接收返回
否
否
是否有组播数据包收到？
是否收到所有的数据包？
否
是
是
是否收到所有的数据包？
否
是
返回到多播监听

图 6